#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_BOOKS 100

#define MAX\_USERS 50

#define MAX\_CART\_ITEMS 10

#define MAX\_ORDERS 50

// Structure to represent a book

typedef struct {

    int id;

    char title[100];

    char author[100];

    float price;

    int quantity;

} Book;

// Structure to represent a user

typedef struct {

    char username[100];

    char password[100];

} User;

// Structure to represent an item in the shopping cart

typedef struct {

    int bookId;

    int quantity;

} CartItem;

// Structure to represent an order

typedef struct {

    int orderId;

    char username[100];

    CartItem items[MAX\_CART\_ITEMS];

    int numItems;

    float totalAmount;

} Order;

// Global arrays to store books, users, shopping carts, and orders

Book books[MAX\_BOOKS];

User users[MAX\_USERS];

CartItem cart[MAX\_CART\_ITEMS];

Order orders[MAX\_ORDERS];

int numBooks = 0;

int numUsers = 0;

int numCartItems = 0;

int numOrders = 0;

// Function prototypes

void createUser();

void displayUsers();

int loginUser();

void addBook();

void displayBooks();

void addToCart();

void viewCart();

void deleteCartItem();

void editBook();

void deleteBook();

void deleteUser();

void placeOrder();

void viewOrders();

int main() {

    int choice;

    char username[100];

    char password[100];

    do {

        printf("\n=== Online Bookstore Management System ===\n");

        printf("1. Create User\n");

        printf("2. Display Users\n");

        printf("3. Login User\n");

        printf("4. Add Book\n");

        printf("5. Display Books\n");

        printf("6. Add to Cart\n");

        printf("7. View Cart\n");

        printf("8. Delete Cart Item\n");

        printf("9. Edit Book\n");

        printf("10. Delete Book\n");

        printf("11. Delete User\n");

        printf("12. Place Order\n");

        printf("13. View Orders\n");

        printf("14. Exit\n");

        printf("Enter your choice: ");

        scanf("%d", &choice);

        switch(choice) {

            case 1:

                createUser();

                break;

            case 2:

                displayUsers();

                break;

            case 3:

                if (loginUser()) {

                    printf("Login successful.\n");

                } else {

                    printf("Login failed. Invalid username or password.\n");

                }

                break;

            case 4:

                addBook();

                break;

            case 5:

                displayBooks();

                break;

            case 6:

                addToCart();

                break;

            case 7:

                viewCart();

                break;

            case 8:

                deleteCartItem();

                break;

            case 9:

                editBook();

                break;

            case 10:

                deleteBook();

                break;

            case 11:

                deleteUser();

                break;

            case 12:

                placeOrder();

                break;

            case 13:

                viewOrders();

                break;

            case 14:

                printf("Exiting...\n");

                exit(0);

            default:

                printf("Invalid choice. Please try again.\n");

        }

    } while (1);

    return 0;

}

// Function to create a new user

void createUser() {

    if (numUsers >= MAX\_USERS) {

        printf("Error: Maximum number of users reached.\n");

        return;

    }

    User newUser;

    printf("Enter username: ");

    scanf("%s", newUser.username);

    printf("Enter password: ");

    scanf("%s", newUser.password);

    users[numUsers++] = newUser;

    printf("User created successfully.\n");

}

// Function to display all users

void displayUsers() {

    if (numUsers == 0) {

        printf("No users available.\n");

        return;

    }

    printf("Users:\n");

    for (int i = 0; i < numUsers; i++) {

        printf("%d. Username: %s\n", i+1, users[i].username);

    }

}

// Function to authenticate a user

int loginUser() {

    char username[100];

    char password[100];

    printf("Enter username: ");

    scanf("%s", username);

    printf("Enter password: ");

    scanf("%s", password);

    for (int i = 0; i < numUsers; i++) {

        if (strcmp(users[i].username, username) == 0 && strcmp(users[i].password, password) == 0) {

            return 1; // User authenticated successfully

        }

    }

    return 0; // Authentication failed

}

// Function to add a new book to the inventory

void addBook() {

    if (numBooks >= MAX\_BOOKS) {

        printf("Error: Maximum number of books reached.\n");

        return;

    }

    Book newBook;

    printf("Enter book title: ");

    scanf("%s", newBook.title);

    printf("Enter author: ");

    scanf("%s", newBook.author);

    printf("Enter price: ");

    scanf("%f", &newBook.price);

    printf("Enter quantity: ");

    scanf("%d", &newBook.quantity);

    newBook.id = numBooks + 1;

    books[numBooks++] = newBook;

    printf("Book added successfully.\n");

}

// Function to display all books in the inventory

void displayBooks() {

    if (numBooks == 0) {

        printf("No books available.\n");

        return;

    }

    printf("ID\tTitle\tAuthor\tPrice\tQuantity\n");

    for (int i = 0; i < numBooks; i++) {

        printf("%d\t%s\t%s\t%.2f\t%d\n", books[i].id, books[i].title, books[i].author, books[i].price, books[i].quantity);

    }

}

// Function to add an item to the shopping cart

void addToCart() {

    if (numUsers == 0) {

        printf("Error: No users registered. Please register first.\n");

        return;

    }

    int bookId, quantity;

    printf("Enter book ID: ");

    scanf("%d", &bookId);

    printf("Enter quantity: ");

    scanf("%d", &quantity);

    if (bookId < 1 || bookId > numBooks) {

        printf("Error: Invalid book ID.\n");

        return;

    }

    if (quantity <= 0) {

        printf("Error: Invalid quantity.\n");

        return;

    }

    if (quantity > books[bookId - 1].quantity) {

        printf("Error: Insufficient quantity in stock.\n");

        return;

    }

    if (numCartItems >= MAX\_CART\_ITEMS) {

        printf("Error: Maximum number of items in the cart reached.\n");

        return;

    }

    CartItem newItem;

    newItem.bookId = bookId;

    newItem.quantity = quantity;

    cart[numCartItems++] = newItem;

    printf("Item added to cart successfully.\n");

}

// Function to display items in the shopping cart

void viewCart() {

    if (numCartItems == 0) {

        printf("Shopping cart is empty.\n");

        return;

    }

    printf("Book\tQuantity\n");

    for (int i = 0; i < numCartItems; i++) {

        printf("%s\t%d\n", books[cart[i].bookId - 1].title, cart[i].quantity);

    }

}

// Function to delete a cart item

void deleteCartItem() {

    if (numCartItems == 0) {

        printf("Shopping cart is empty.\n");

        return;

    }

    int index;

    printf("Enter index of item to delete: ");

    scanf("%d", &index);

    if (index < 1 || index > numCartItems) {

        printf("Error: Invalid index.\n");

        return;

    }

    // Shift items to fill the gap

    for (int i = index - 1; i < numCartItems - 1; i++) {

        cart[i] = cart[i + 1];

    }

    numCartItems--;

    printf("Item deleted from cart successfully.\n");

}

// Function to edit book details

void editBook() {

    if (numBooks == 0) {

        printf("No books available.\n");

        return;

    }

    int bookId;

    printf("Enter ID of book to edit: ");

    scanf("%d", &bookId);

    if (bookId < 1 || bookId > numBooks) {

        printf("Error: Invalid book ID.\n");

        return;

    }

    printf("Enter new title: ");

    scanf("%s", books[bookId - 1].title);

    printf("Enter new author: ");

    scanf("%s", books[bookId - 1].author);

    printf("Enter new price: ");

    scanf("%f", &books[bookId - 1].price);

    printf("Enter new quantity: ");

    scanf("%d", &books[bookId - 1].quantity);

    printf("Book details updated successfully.\n");

}

// Function to delete a book

void deleteBook() {

    if (numBooks == 0) {

        printf("No books available.\n");

        return;

    }

    int bookId;

    printf("Enter ID of book to delete: ");

    scanf("%d", &bookId);

    if (bookId < 1 || bookId > numBooks) {

        printf("Error: Invalid book ID.\n");

        return;

    }

    // Shift books to fill the gap

    for (int i = bookId - 1; i < numBooks - 1; i++) {

        books[i] = books[i + 1];

    }

    numBooks--;

    printf("Book deleted successfully.\n");

}

// Function to delete a user

void deleteUser() {

    if (numUsers == 0) {

        printf("No users available.\n");

        return;

    }

    int userId;

    printf("Enter ID of user to delete: ");

    scanf("%d", &userId);

    if (userId < 1 || userId > numUsers) {

        printf("Error: Invalid user ID.\n");

        return;

    }

    // Shift users to fill the gap

    for (int i = userId - 1; i < numUsers - 1; i++) {

        users[i] = users[i + 1];

    }

    numUsers--;

    printf("User deleted successfully.\n");

}

// Function to place an order

void placeOrder() {

    if (numUsers == 0) {

        printf("Error: No users registered. Please register first.\n");

        return;

    }

    if (numCartItems == 0) {

        printf("Error: Shopping cart is empty. Please add items to cart first.\n");

        return;

    }

    Order newOrder;

    newOrder.orderId = numOrders + 1;

    strcpy(newOrder.username, users[numUsers - 1].username); // Place order for the last logged-in user

    newOrder.numItems = numCartItems;

    newOrder.totalAmount = 0;

    for (int i = 0; i < numCartItems; i++) {

        int bookId = cart[i].bookId;

        int quantity = cart[i].quantity;

        newOrder.items[i].bookId = bookId;

        newOrder.items[i].quantity = quantity;

        // Update total amount

        newOrder.totalAmount += books[bookId - 1].price \* quantity;

        // Update book quantity

        books[bookId - 1].quantity -= quantity;

    }

    orders[numOrders++] = newOrder;

    // Clear the cart

    numCartItems = 0;

    printf("Order placed successfully. Total amount: %.2f\n", newOrder.totalAmount);

}

// Function to view all orders

void viewOrders() {

    if (numOrders == 0) {

        printf("No orders available.\n");

        return;

    }

    printf("Orders:\n");

    printf("OrderID\tUsername\tTotalAmount\n");

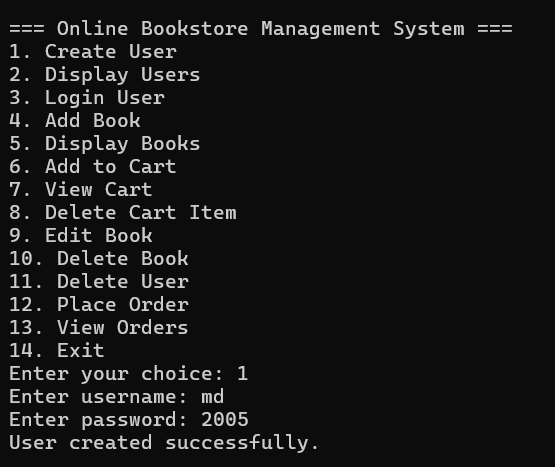
    for (int i = 0; i < numOrders; i++) {

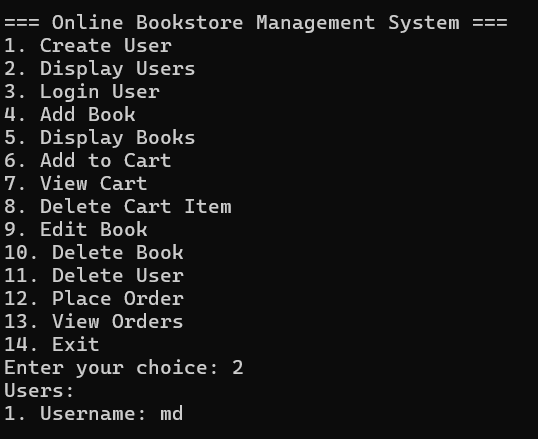
        printf("%d\t%s\t%.2f\n", orders[i].orderId, orders[i].username, orders[i].totalAmount);

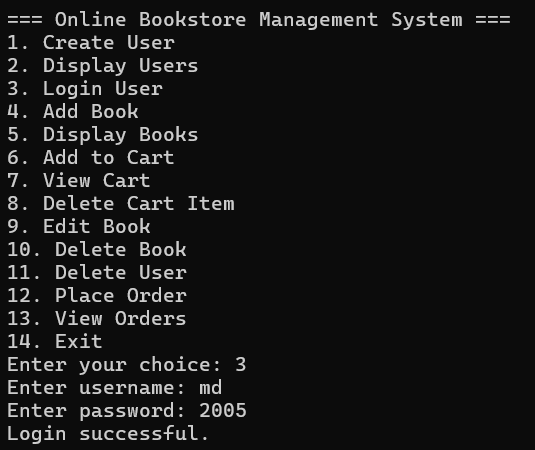
    }

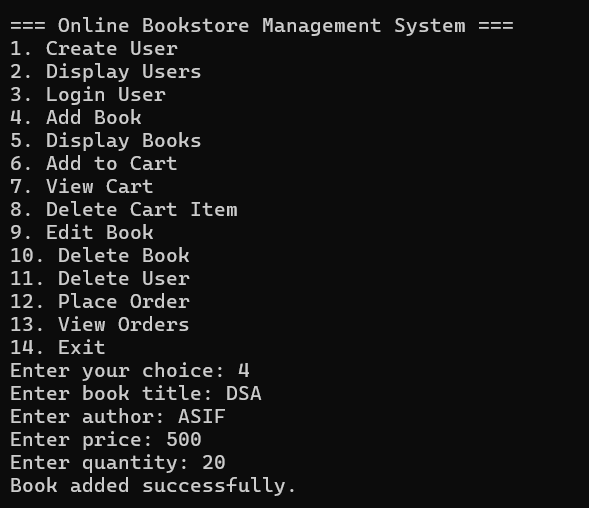
}

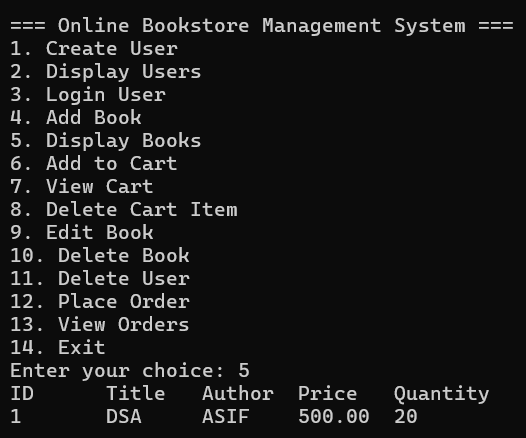
**OUTPUT**

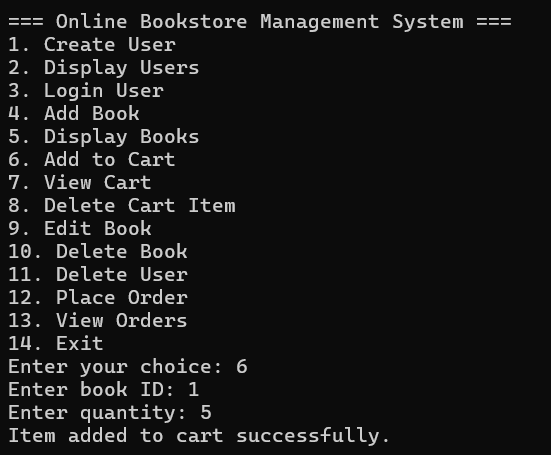
****

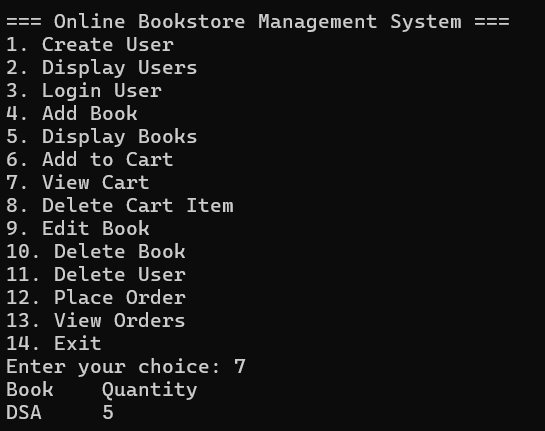
****

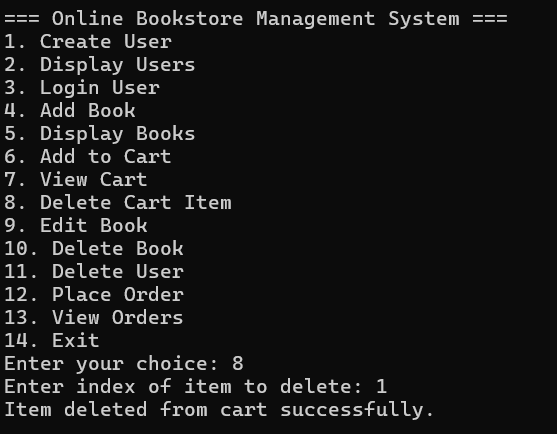
****

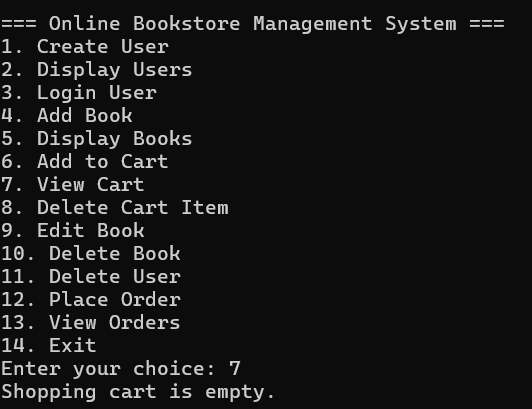
****

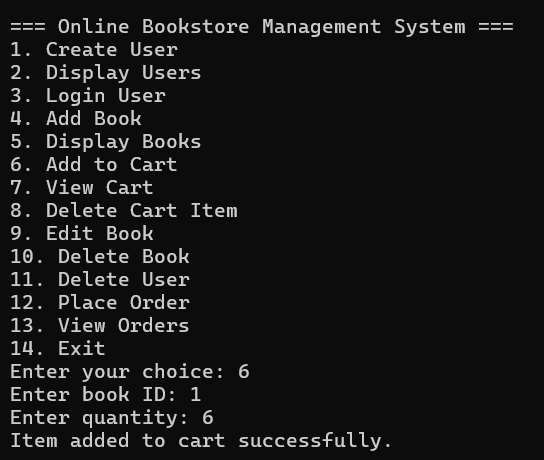
****

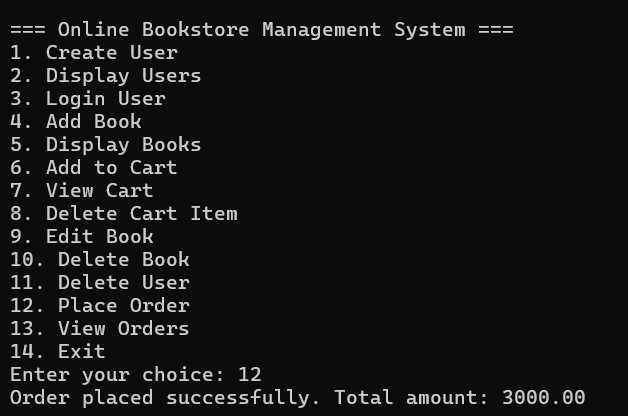
****

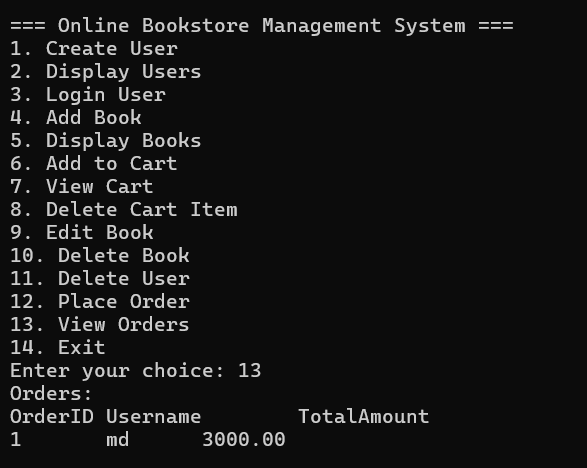
****

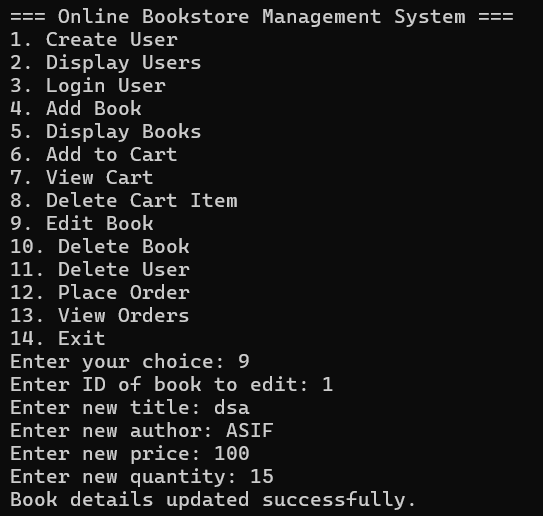
****

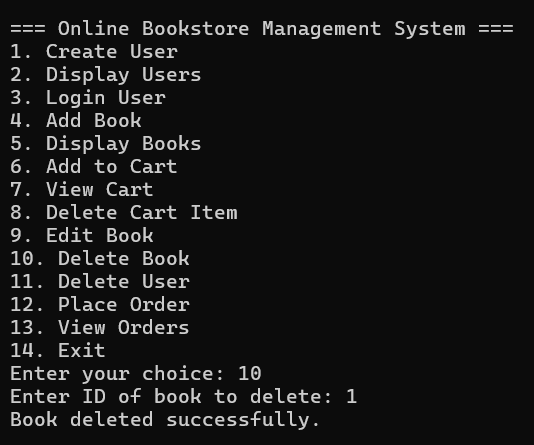
****

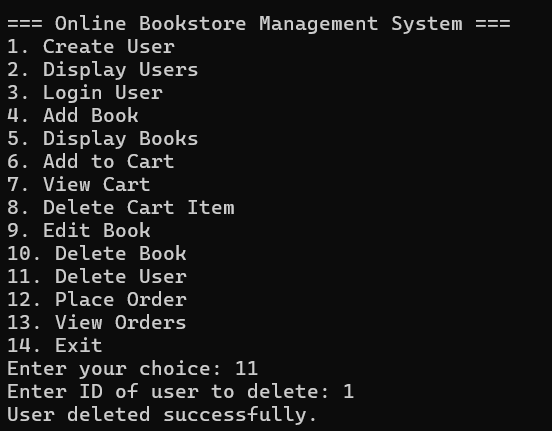
****

****

****

****

****

****